Pattern Alternating Maximization Algorithm for High-Dimensional Missing Data

Nicolas Städler
Seminar für Statistik, ETH Zürich
CH-8092 Zürich, Switzerland.
staedler@stat.math.ethz.ch

Peter Bühlmann
Seminar für Statistik, ETH Zürich
CH-8092 Zürich, Switzerland.
buhlmann@stat.math.ethz.ch

July 12, 2010

Abstract

We propose a new and computationally efficient algorithm for maximizing the observed log-likelihood for a multivariate normal data matrix with missing values. We show that our procedure based on iteratively regressing the missing on the observed variables, generalizes the traditional EM algorithm by alternating between different complete data spaces and performing the E-Step incrementally. In this non-standard setup we prove numerical convergence to a stationary point of the observed log-likelihood.

For high-dimensional data, where the number of variables may greatly exceed sample size, we add a Lasso penalty in the regression part of our algorithm and perform coordinate descent approximations. This leads to a computationally very attractive technique with sparse regression coefficients for missing data imputation. Simulations and results on four microarray datasets show that the new method often outperforms alternative imputation techniques as k-nearest neighbors imputation, nuclear norm minimization or a penalized likelihood approach with an ℓ_1-penalty on the inverse covariance matrix.

Keywords Missing data, observed likelihood, (partial) E- and M-Step, Lasso, matrix completion

1 Introduction and Motivation

Missing data imputation for large datasets is an essential pre-processing step in complex data applications. A well-known example are microarray datasets which contain the expression values of thousands of genes from a series of experiments. Missing values are inherent to such datasets. They occur for diverse reasons, e.g. insufficient resolution, image corruption, dust or scratches on the slides. Apart from microarrays, much attention has been recently given to the so-called matrix completion problem. Most prominent in this context is the "Netflix" movie rating dataset with rows corresponding to customers and columns representing their movie rating and the customers have seen/rated only a small fraction of all possible movies. The goal is to estimate the ratings for unseen movies. In this paper, we propose a new and computationally efficient EM-type algorithm for missing value imputation in the high-dimensional multivariate normal model where the number of variables p can greatly exceed the number of independent samples n. While our motivating examples are microarray datasets with expression values of p different genes (variables) and n different experimental conditions (samples), our method is applicable to a wide range of problems where the data is arranged in the form of a $n \times p$ matrix with a possibly large fraction of the entries missing. The Gaussian assumption in our model is used for computation of the likelihood but empirical findings suggest that the method is useful for many continuous data matrices.

There is a growing literature of missing value imputation methods. For microarray data, examples are k-nearest neighbors imputation and singular value decomposition imputation (Troyanskaya *et al.*, 2001), imputation based on Bayesian principal component analysis (Oba *et al.*, 2003) or the local least squares imputation from Kim *et al.* (2006). For a review and a discussion on microarray data imputation see Aittokallio (2010). In the context of the so-called matrix completion problem, where the goal is to recover a low-rank matrix from an incomplete set of entries, it has been shown in a series of fascinating papers that one can recover the missing data entries by solving a convex optimization problem, namely, nuclear-norm minimization subject to data constraints (Candès and Recht, 2009; Candès and Tao, 2009; Keshavan *et al.*, 2009). Efficient convex algorithms for the matrix completion problem were proposed by Cai *et al.* (2008) and Mazumder *et al.* (2009). If the missing data problem does not arise from a near low rank matrix scenario, there is substantial room to improve upon the convex matrix completion algorithms. We will empirically demonstrate this point for some microarray high-throughput biological data.

Here, we address the missing data problem through a likelihood approach (Little and Rubin, 1987; Schafer, 1997). We model the correlation between different variables in the data by using the Multivariate Normal Model (MVN) $\mathcal{N}(\mu, \Sigma)$ with a p-dimensional covariance matrix Σ. Recently, in the high-dimensional setup with $p \gg n$, Städler and Bühlmann (2010) proposed to maximize the penalized observed log-likelihood with an ℓ_1-penalty on the inverse covariance matrix. They called their method the *MissGLasso*, as an extension of the GLasso (Friedman *et al.*, 2007) for missing data. A similar approach, in the context of so-called transposable models, is given by Allen and Tibshirani (2010).

The *MissGLasso* uses an EM algorithm for optimization of the penalized observed log-likelihood. Roughly, the algorithm can be summarized as follows. In the E-Step, for

each sample, the regression coefficients of the missing against the observed variables are computed from the current estimated covariance matrix $\hat{\Sigma}$. In the following M-Step, the missing values are imputed by linear regressions and $\hat{\Sigma}$ is re-estimated by performing a GLasso on completed data. There are two main drawbacks of this algorithm in a high-dimensional context. First, the E-Step is rather complex as it involves (for each sample) inversion and multiplication of large matrices in order to compute the regression coefficients. Secondly, a sparse inverse covariance does not imply sparse regression coefficients while we believe that in high-dimensions, sparse regression coefficients would enhance imputations.

Our new algorithm, *MissPALasso* (Missingness Pattern Alternating Lasso algorithm) in this paper, generalizes the E-Step in order to resolve the disadvantages of the *MissGLasso*. In particular, inversion of a matrix (in order to compute the regression coefficients) will be replaced by a simple soft-thresholding operator. In addition, the regression coefficients will be sparse, which leads to a new sparsity concept for missing data estimation.

In order to motivate the *MissPALasso*, we develop first the Missingness Pattern Alternating maximization algorithm (*MissPA*) for optimizing the (unpenalized) observed log-likelihood. The *MissPA* generalizes the E- *and* M-Step of the traditional EM originally proposed by Dempster *et al.* (1977) by alternating between different complete data spaces *and* performing the E-Step incrementally. Such a generalization does not fit into any of the existing methodologies which extend the standard EM. However, by exploiting the special structure of our procedure and applying standard properties of the Kullback-Leibler divergence, we prove convergence to a stationary point of the observed log-likelihood.

The further organization of the paper is as follows: Section 2 introduces the setup and the useful notion of missingness patterns. In Section 3 we develop our algorithms: the pattern alternating maximization algorithm (*MissPA*) is presented in Section 3.2 and the *MissPALasso* then follows in Section 3.3 as an adapted version for high-dimensional data with $p \gg n$. Section 4 compares performance of *MissPALasso* with other imputation methods on simulated and real data and reports on computational efficiency. Finally, in Section 5, we present some mathematical theory which describes the numerical properties of the pattern alternating maximization algorithm.

2 Setup

We assume $X = (X_1, \ldots, X_p) \sim \mathcal{N}(\mu, \Sigma)$ has a p-variate normal distribution with mean μ and covariance Σ. In order to simplify the notation we set without loss of generality $\mu = 0$: for $\mu \neq 0$, some of the formulae involve the parameter μ and an intercept column of $(1, \ldots, 1)$ in the design matrices but conceptually, we can proceed as for the case with $\mu = 0$. We then write $\mathbf{X} = (\mathbf{X}_{\text{obs}}, \mathbf{X}_{\text{mis}})$, where $\mathbf{X}$ represents an i.i.d. random sample of size n, $\mathbf{X}_{\text{obs}}$ denotes the set of observed values, and $\mathbf{X}_{\text{mis}}$ the missing data.

Missingness patterns and different parametrizations For our purpose it will be convenient to group rows of the matrix $\mathbf{X}$ according to their missingness patterns (Schafer,

1997). We index the unique missingness patterns that actually appear in our data by $k = 1, \ldots, s$. Furthermore, with $o_k \subset \{1, \ldots, p\}$ and $m_k = \{1, \ldots, p\} \setminus o_k$ we denote the set of observed variables and the set of missing variables, respectively. $\mathcal{I}_k$ is the index set of the samples (row numbers) which belong to pattern k, whereas $\mathcal{I}_k^c = \{1, \ldots, n\} \setminus \mathcal{I}_k$ stands for the row numbers which do not belong to that pattern. By convention, samples with all variables observed do not belong to a missingness pattern.

Consider a partition $X = (X_{o_k}, X_{m_k})$ of a single Gaussian random vector. It's well known that $X_{m_k}|X_{o_k}$ follows a linear regression on X_{o_k} with regression coefficients $B_{m_k|o_k}$ and covariance $\Sigma_{m_k|o_k}$ given by

$$\begin{aligned} B_{m_k|o_k} &= \Sigma_{m_k,o_k}\Sigma_{o_k}^{-1}, \\ \Sigma_{m_k|o_k} &= \Sigma_{m_k} - \Sigma_{m_k,o_k}\Sigma_{o_k}^{-1}\Sigma_{o_k,m_k}. \end{aligned} \tag{2.1}$$

Consequently, we can write the density $p(x; \Sigma)$ of X as

$$p(x; \Sigma) = p(x_{m_k}|x_{o_k}; B_{m_k|o_k}, \Sigma_{m_k|o_k})p(x_{o_k}; \Sigma_{o_k}),$$

i.e., the density can be characterized by either the parameter Σ or $(\Sigma_{o_k}, B_{m_k|o_k}, \Sigma_{m_k|o_k})$. With the transformation (2.1) we can switch between both parametrizations.

Observed log-likelihood and Maximum Likelihood Estimation (MLE) A systematic approach to estimate the parameter of interest Σ from $\mathbf{X}_{\text{obs}}$ maximizes the observed log-likelihood $\ell(\Sigma; \mathbf{X}_{\text{obs}})$ given by

$$\ell(\Sigma; \mathbf{X}_{\text{obs}}) = \sum_{i \notin \bigcup_k \mathcal{I}_k} \log p(x_i; \Sigma) + \sum_{k=1}^{s} \sum_{i \in \mathcal{I}_k} \log p(x_{i,o_k}; \Sigma_{o_k}). \tag{2.2}$$

Inference for Σ can be based on the observed log-likelihood (2.2) if the underlying missing data mechanism is *ignorable*. The missing data mechanism is said to be *ignorable* if the probability that an observation is missing may depend on $\mathbf{X}_{\text{obs}}$ but not on $\mathbf{X}_{\text{mis}}$ (*Missing at Random*) and if the parameters of the data model and the parameters of the missingness mechanism are *distinct*. For a precise definition see Little and Rubin (1987).

Explicit maximization of $\ell(\Sigma; \mathbf{X}_{\text{obs}})$ is only possible for special missing data patterns. Most prominent are examples with a so-called monotone missing data pattern (Little and Rubin, 1987; Schafer, 1997), where X_1 is more observed than X_2, which is more observed than X_3, and so on. In this case, the observed log-likelihood factorizes and explicit maximization is achieved by performing several regressions. For a general pattern of missing data, the traditional EM algorithm is often used for optimization of (2.2). See Schafer (1997) for a detailed description of the algorithm. In the next section we present an alternative method for maximizing the observed log-likelihood. We will argue that this new algorithm is computationally more efficient than the traditional EM.

3 Pattern Alternating Missing Data Estimation and ℓ_1- regularization

For each missingness pattern, indexed by $k = 1, \ldots, s$, we introduce some further notation:

$$\begin{aligned}\mathbf{X}^k &= (x_{i,j}) \quad \text{with} \quad i \in \mathcal{I}_k \quad j = 1, \ldots, p \\ \mathbf{X}^{-k} &= (x_{i,j}) \quad \text{with} \quad i \in \mathcal{I}_k^c \quad j = 1, \ldots, p.\end{aligned}$$

Thus, $\mathbf{X}^k$ is the $|\mathcal{I}_k| \times p$ submatrix of $\mathbf{X}$ with rows belonging to the kth pattern. Similarly, $\mathbf{X}^{-k}$ is the $|\mathcal{I}_k^c| \times p$ matrix with rows not belonging to the kth pattern. In the same way we define $\mathbf{X}^k_{o_k}, \mathbf{X}^k_{m_k}, \mathbf{X}^{-k}_{o_k}$ and $\mathbf{X}^{-k}_{o_k}$. For example, $\mathbf{X}^k_{o_k}$ is defined as the $|\mathcal{I}_k| \times |o_k|$ matrix with

$$\mathbf{X}^k_{o_k} = (x_{i,j}) \quad \text{with} \quad i \in \mathcal{I}_k, \quad j \in o_k.$$

3.1 MLE for Data with a Single Missingness Pattern

Assume that the data matrix $\mathbf{X}$ has only one single missingness pattern, denoted by s. This is the most simple example of a monotone pattern. The observed log-likelihood factorizes according to:

$$\begin{aligned}\ell(\Sigma; \mathbf{X}_{\text{obs}}) &= \sum_{i \in \mathcal{I}_s} \log p(x_{i,o_s}; \Sigma_{o_s}) + \sum_{i \in \mathcal{I}_s^c} \log p(x_i; \Sigma) \\ &= \sum_{i=1}^{n} \log p(x_{i,o_s}; \Sigma_{o_s}) + \sum_{i \in \mathcal{I}_s^c} \log p(x_{i,m_s} | x_{i,o_s}; B_{m_s|o_s}, \Sigma_{m_s|o_s}). \qquad (3.3)\end{aligned}$$

The left and right part in Equation (3.3) can be maximized separately. The first part is maximized by the sample covariance of the observed variables based on *all samples*, whereas the second part is maximized by a regression of the missing against observed variables based on only the *fully observed samples*. In formulae:

$$\hat{\Sigma}_{o_s} = {}^t\mathbf{X}_{o_s}\mathbf{X}_{o_s}/n, \qquad (3.4)$$

and

$$\begin{aligned}\hat{B}_{m_s|o_s} &= {}^t\mathbf{X}^{-s}_{m_s}\mathbf{X}^{-s}_{o_s}({}^t\mathbf{X}^{-s}_{o_s}\mathbf{X}^{-s}_{o_s})^{-1}, \\ \hat{\Sigma}_{m_s|o_s} &= {}^t(\mathbf{X}^{-s}_{m_s} - \mathbf{X}^{-s}_{o_s}\,{}^t\hat{B}_{m_s|o_s})(\mathbf{X}^{-s}_{m_s} - \mathbf{X}^{-s}_{o_s}\,{}^t\hat{B}_{m_s|o_s})/|\mathcal{I}_s^c|. \qquad (3.5)\end{aligned}$$

Having these estimates at hand, it is easy to impute the missing data:

$$\hat{x}_{i,m_s} = \hat{B}_{m_s|o_s}\,{}^t x_{i,o_s} \text{ for all } i \in \mathcal{I}_s, \quad \text{or, in matrix notation,} \quad \hat{\mathbf{X}}^s_{m_s} = \mathbf{X}^s_{o_s}\,{}^t\hat{B}_{m_s|o_s}.$$

It is important to note, that, if interested in imputation, only the regression part of the MLE is needed and the estimate $\hat{\Sigma}_{o_s}$ in (3.4) is superfluous.

3.2 MLE for General Missing Data Pattern

We turn now to the general case, where we have more than one missingness pattern, indexed by $k = 1, \ldots, s$. The general idea of the new algorithm is as follows. Assume we have some initial imputations for all missing values. Our goal is to improve on these imputations. For this purpose, we iterate as follows:

- Keep all imputations except those of the 1st missingness pattern fixed and compute the single pattern MLE (for the first pattern) as explained in Section 3.1. In particular, compute the regression coefficients of the missing 1st pattern against all other variables (treated as "observed").
- Use the resulting estimates (regression coefficients) to impute the missing values from only the 1st pattern.

Next, turn to the 2nd pattern and repeat the above steps. In this way we continue cycling through the different patterns until convergence.

We now describe the Pattern Alternating Maximization algorithm which makes the above idea precise. Let $T = {}^t\mathbf{X}\mathbf{X}$ be the sufficient statistic in the multivariate normal model. Furthermore, denote by $T^k = {}^t(\mathbf{X}^k)\mathbf{X}^k$, $T^{-k} = {}^t(\mathbf{X}^{-k})\mathbf{X}^{-k} = \sum_{l \neq k} T^l$. Let $\mathcal{T}$ and $\mathcal{T}^k\,(k = 1, \ldots, s)$ be some initial guess of T and $T^k\,(k = 1, \ldots, s)$, for example, using zero imputation. Our algorithm proceeds as follows.

Missingness Pattern Alternating Maximization Algorithm (MissPA)

---

(1) $\mathcal{T}$, $\mathcal{T}^k$: initial guess of T and T^k $(k = 1, \ldots, s)$.

(2) For $k = 1, \ldots, s$ do:

M-Step: Compute the MLE $\hat{B}_{m_k|o_k}$, and $\hat{\Sigma}_{m_k|o_k}$, based on $\mathcal{T}^{-k} = \mathcal{T} - \mathcal{T}^k$:

$$\hat{B}_{m_k|o_k} = \mathcal{T}^{-k}_{m_k,o_k}(\mathcal{T}^{-k}_{o_k,o_k})^{-1},$$
$$\hat{\Sigma}_{m_k|o_k} = \left(\mathcal{T}^{-k}_{m_k,m_k} - \mathcal{T}^{-k}_{m_k,o_k}(\mathcal{T}^{-k}_{o_k,o_k})^{-1}\mathcal{T}^{-k}_{o_k,m_k}\right)/|\mathcal{I}^c_k|.$$

Partial E-Step:

Set $\mathcal{T}^k = \mathbb{E}[T^k|\mathbf{X}^k_{o_k}, \hat{B}_{m_k|o_k}, \hat{\Sigma}_{m_k|o_k}]$,
Update $\mathcal{T} = \mathcal{T}^{-k} + \mathcal{T}^k$.

(3) Repeat step (2) until some convergence criterion is met.

(4) Compute the final maximum likelihood estimator $\hat{\Sigma}$ via:

$$\hat{\Sigma}_{o_s} = \mathcal{T}_{o_s,o_s}/n,\ \hat{\Sigma}_{m_s,o_s} = \hat{B}_{m_s|o_s}\hat{\Sigma}_{o_s} \text{ and } \hat{\Sigma}_{m_s} = \hat{\Sigma}_{m_s|o_s} + \hat{B}_{m_s|o_s}\hat{\Sigma}_{o_s,m_s}.$$

Note, that we refer to the maximization step as M-Step and to the imputation step as *partial* E-Step. The word partial refers to the fact that the expectation is only performed with respect to samples belonging to the current pattern. The partial E-Step of our algorithm takes the following simple form:

$$\mathcal{T}^k_{o_k,m_k} = {}^t(\mathbf{X}^k_{o_k})\hat{\mathbf{X}}^k_{m_k},$$
$$\mathcal{T}^k_{m_k,m_k} = {}^t(\hat{\mathbf{X}}^k_{m_k})\hat{\mathbf{X}}^k_{m_k} + |\mathcal{I}_k|\hat{\Sigma}_{m_k|o_k},$$

where $\hat{\mathbf{X}}^k_{m_k} = \mathbb{E}[\mathbf{X}^k_{m_k}|\mathbf{X}^k_{o_k}, \hat{B}_{m_k|o_k}, \hat{\Sigma}_{m_k|o_k}] = \mathbf{X}^k_{o_k}\,{}^t\hat{B}_{m_k|o_k}$.

Our algorithm does not require an evaluation of $\hat{\Sigma}_{o_k}$ in the M-Step, as it is not used in the following partial E-Step. But, if we are interested in the observed log-likelihood or the maximum likelihood estimator $\hat{\Sigma}$ at convergence, we compute $\hat{\Sigma}_{o_s}$ (at convergence), use it together with $\hat{B}_{m_s|o_s}$ and $\hat{\Sigma}_{m_s|o_s}$ to get $\hat{\Sigma}$ via the transformations (2.1) as explained in step (4).

The MissPA is computationally more efficient than the traditional EM for missing data: one cycle through all patterns ($k = 1, \ldots, s$) takes about the same time than one iteration of a usual EM. But our algorithm makes more progress since the information from the partial E-Step is utilized immediately to perform the next M-Step. We will demonstrate empirically the gain of computational efficiency in Section 4.2.

The new MissPA generalizes the traditional EM in two ways. First, the MissPA alternates between different complete data spaces in the sense of the SAGE algorithm of Fessler and Hero (1994). Secondly, the E-Step is performed incrementally (Neal and Hinton, 1998).

There has been no framework for both of these generalizations. In Section 5.2 we will introduce an appropriate framework for proving numerical convergence to stationary points of the observed log-likelihood $\ell(\Sigma; \mathbf{X}_{\text{obs}})$.

Remark 3.1. *A slight modification of the MissPA, namely replacing everywhere in the M-Step $\mathcal{T}^{-k}$ by $\mathcal{T}$ results in an alternative algorithm which can be viewed as an incremental EM in the sense of Neal and Hinton (1998). Furthermore, we can adapt the partial E-Step using the updating rule $\mathcal{T} = \gamma\mathcal{T} + \mathcal{T}^k$, with $\gamma = 1 - |\mathcal{I}_k|/n$. Here, it is important to update $\mathcal{T}$ also with respect to the fully observed observations. Note also, that we only need to save $\mathcal{T}$, whereas the MissPA also requires $\mathcal{T}^k$, for $k = 1, \ldots, s$. However, the described modified algorithm is typically inferior to the MissPA.*

3.3 High-Dimensionality, Lasso Penalty and a Coordinate Descent Approximation

It is clear that in a high-dimensional framework with $p \gg n$ some regularization is necessary. In the M-Step of the MissPA we solve a multivariate regression problem. The main idea is, in order to regularize, to replace the regressions with a Lasso (Tibshirani, 1996). We give now the details.

Estimation of $B_{m_k|o_k}$: The estimation of the multivariate regression coefficients in the M-Step of the MissPA can be expressed as $|m_k|$ separate minimization problems of the form

$$\hat{B}_{j|o_k} = \arg\min_{\beta} -\mathcal{T}^{-k}_{j,o_k}\beta + {}^t\beta\mathcal{T}^{-k}_{o_k,o_k}\beta/2,$$

where $j \in m_k$. Here, $\hat{B}_{j|o_k}$ denotes the jth row vector of the $(|m_k| \times |o_k|)$-matrix $\hat{B}_{m_k|o_k}$ and represents the regression of variable j against the variables from o_k.

Consider now the objective function

$$-\mathcal{T}^{-k}_{j,o_k}\beta + {}^t\beta\mathcal{T}^{-k}_{o_k,o_k}\beta/2 + \lambda\|\beta\|_1, \tag{3.6}$$

with an additional Lasso penalty. Instead of minimizing (3.6) with respect to β (for all $j \in m_k$), it is computationally much more efficient to improve it coordinate-wise only from the old parameters (computed in the last cycle through all patterns). For that purpose, let $B^{(r)}_{m_k|o_k}$ be the regression coefficients for pattern k in cycle r and $B^{(r)}_{j|o_k}$ its jth row vector. In cycle $r+1$ we compute $B^{(r+1)}_{j|o_k}$ by minimizing (3.6) with respect to each of the components of β, holding the other components fixed at their current value.

Closed-form updates have the form:

$$B^{(r+1)}_{j|l} = \frac{Soft\left(\mathcal{T}^{-k}_{l,l}B^{(r)}_{j|l} - S_l, \lambda\right)}{\mathcal{T}^{-k}_{l,l}}, \qquad \text{for all } l \in o_k, \tag{3.7}$$

where

- $B^{(r+1)}_{j|l}$ is the lth component of $B^{(r+1)}_{j|o_k}$ equal to the element (j,l) of matrix $B^{(r+1)}_{m_k|o_k}$.
- S_l, the gradient of $-\mathcal{T}^{-k}_{j,o_k}\beta + {}^t\beta\mathcal{T}^{-k}_{o_k,o_k}\beta/2$ with respect to β_l, which equals

$$S_l = -\mathcal{T}^{-k}_{j,l} + \sum_{\substack{v<l \\ v\in o_k}} \mathcal{T}^{-k}_{l,v}B^{(r+1)}_{j|v} + \mathcal{T}^{-k}_{l,l}B^{(r)}_{j|l} + \sum_{\substack{v>l \\ v\in o_k}} \mathcal{T}^{-k}_{l,v}B^{(r)}_{j|v}. \tag{3.8}$$

- $Soft(z,\lambda) = \begin{cases} z-\lambda & \text{if } z > \lambda \\ z+\lambda & \text{if } z < -\lambda \\ 0 & \text{if } |z| \le \lambda \end{cases}$, is the standard soft-thresholding operator.

In a sparse setup the soft-thresholding update (3.7) can be evaluated very quickly as l varies and often coefficients which are zero remain zero after thresholding. See also the *naive-* or *covariance update* idea of Friedman *et al.* (2008) for efficient computation of (3.7) and (3.8).

Estimation of $\Sigma_{m_k|o_k}$: We update the residual covariance matrix as:

$$\Sigma^{(r+1)}_{m_k|o_k} = \left(\mathcal{T}^{-k}_{m_k,m_k} - \mathcal{T}^{-k}_{m_k,o_k}{}^tB^{(r+1)}_{m_k|o_k} - B^{(r+1)}_{m_k|o_k}\mathcal{T}^{-k}_{o_k,m_k} + B^{(r+1)}_{m_k|o_k}\mathcal{T}^{-k}_{o_k,o_k}{}^tB^{(r+1)}_{m_k|o_k}\right)/|\mathcal{I}^c_k|. \tag{3.9}$$

Formula (3.9) can be viewed as a generalized version of Equation (3.5), when multiplying out the matrix product in (3.5) and taking expectations.

Our regularized algorithm, the MissPALasso, is summarized in Table 1. Note, that the partial E-Step remains the same as for the MissPA of Section 3.2. As we are mainly interested in estimating the missing values, we will output the data matrix with missing values imputed by the regression coefficients $\hat{B}_{m_k|o_k}$ $(k = 1,\ldots,s)$ as indicated in step (4)

of Table 1. The MissPALasso provides not only the imputed data matrix $\hat{\mathbf{X}}$ but also $\hat{\mathcal{T}}$, the completed version of the sufficient statistic ${}^t\mathbf{X}\mathbf{X}$. The latter can be very useful if the MissPALasso is used as a pre-processing step followed by a learning method which is expressible in terms of the sufficient statistic. Examples include regularized regression (e.g., Lasso), discriminant analysis, or estimation of directed acyclic graphs with the PC-algorithm (Spirtes *et al.*, 2000).

By construction, the regression estimates $\hat{B}_{m_k|o_k}$ are sparse, due to the employed ℓ_1-penalty, and therefore, imputation of missing values $\hat{\mathbf{X}}^k_{m_k} = \mathbf{X}^k_{o_k}\,{}^t\hat{B}_{m_k|o_k}$ is based on sparse regressions. This is in sharp contrast to the MissGLasso approach (see Section 4.1) which places sparsity on Σ^{-1}. But this does not imply that regressions of variables in m_k on variables in o_k are sparse since the inverse of sub-matrices of a sparse Σ^{-1} are not sparse in general. The MissPALasso employs another type of sparsity and this seems to be the main reason for its better statistical performance than MissGLasso.

Remark 3.2. *In practice, we propose to compute the MissPALasso for a decreasing sequence of values for λ, using each solution as a warm start for the next problem with smaller λ. This pathwise strategy is computationally very attractive and our algorithm converges (for each λ) after a few cycles.*

Missingness Pattern Alternating Imputation and ℓ_1-Penalty (MissPALasso)

(1) Set $r = 0$ and start with initial guess for $\mathcal{T}$, $\mathcal{T}^k$ and $B^{(0)}_{m_k|o_k}$ $(k = 1, \ldots, s)$.

(2) In cycle $r + 1$; for $k = 1, \ldots, s$ do:

M-Step:

For all $j \in m_k$, compute $B^{(r+1)}_{j|o_k}$ by improving $-\mathcal{T}^{-k}_{j,o_k}\beta + {}^t\beta\mathcal{T}^{-k}_{o_k,o_k}\beta/2 + \lambda||\beta||_1$ in a coordinate-wise manner from $B^{(r)}_{j|o_k}$.

Set $\Sigma^{(r+1)}_{m_k|o_k} = \left(\mathcal{T}^{-k}_{m_k,m_k} - \mathcal{T}^{-k}_{m_k,o_k}\,{}^tB^{(r+1)}_{m_k|o_k} - B^{(r+1)}_{m_k|o_k}\mathcal{T}^{-k}_{o_k,m_k} + B^{(r+1)}_{m_k|o_k}\mathcal{T}^{-k}_{o_k,o_k}\,{}^tB^{(r+1)}_{m_k|o_k}\right)/|\mathcal{I}^c_k|$.

Partial E-Step:

Set $\mathcal{T}^k = \mathbb{E}[T^k|\mathbf{X}^k_{o_k}, B^{(r+1)}_{m_k|o_k}, \Sigma^{(r+1)}_{m_k|o_k}]$,

Update $\mathcal{T} = \mathcal{T}^{-k} + \mathcal{T}^k$.

Increase: $r \leftarrow r + 1$.

(3) Repeat step (2) until some convergence criterion is met.

(4) Output the imputed data matrix $\hat{\mathbf{X}}$, with missing values estimated by: $\hat{\mathbf{X}}^k_{m_k} = \mathbf{X}^k_{o_k}\,{}^t\hat{B}_{m_k|o_k}$, $k = 1, \ldots, s$.

Table 1: MissPALasso. In the kth M-Step of cycle $r + 1$, instead of a multivariate Lasso regression, a coordinate descent approximation of the corresponding Lasso problem is performed.

4 Numerical Experiments

4.1 Performance of the MissPALasso

In this section we will explore the performance of the MissPALasso developed in Section 3.3. We compare our new method with alternative ways of imputing missing values in high-dimensional data.

We consider the following methods:

- *KnnImpute:* Impute the missing values by the K-nearest neighbors imputation method introduced by Troyanskaya *et al.* (2001).
- *SoftImpute:* The soft imputation algorithm is proposed by Mazumder *et al.* (2009) in order to solve the matrix completion problem. They propose to approximate the incomplete data matrix $\mathbf{X}$ by a complete (low-rank) matrix $\mathbf{Z}$ minimizing

$$\frac{1}{2}\sum_{(i,j)\in\Omega}(z_{ij}-x_{ij})^2+\lambda\|\mathbf{Z}\|_*.$$

  Here, Ω denotes the indices of observed entries and $\|\mathbf{Z}\|_*$ is the nuclear norm, or the sum of the singular values. The missing values of $\mathbf{X}$ are imputed by the corresponding values of $\mathbf{Z}$.
- *MissGLasso:* Compute $\hat{\Sigma}$ by minimizing $-\ell(\Sigma;\mathbf{X}_{obs})+\lambda\|\Sigma^{-1}\|_1$, where $\|\Sigma^{-1}\|_1$ is the entrywise ℓ_1-norm. Then, use this estimate to impute the missing values by conditional mean imputation. The *MissGLasso* is described in Städler and Bühlmann (2010).
- *MissPALasso:* This is the method introduced in Section 3.3.

All methods involve one tuning parameter. In KnnImpute we have to choose the number K of nearest neighbors, while SoftImpute, MissGLasso and MissPALasso involve a regularization parameter which is always denoted by λ. In all of our experiments we choose the tuning parameters to obtain optimal performance for imputation.

To assess the performances of the methods we use the normalized root mean squared error (Oba *et al.*, 2003) which is defined by

$$\text{NRMSE} \;=\; \sqrt{\frac{\mathbf{mean}\left((\mathbf{X}^{\text{true}}-\hat{\mathbf{X}})^2\right)}{\mathbf{var}\left(\mathbf{X}^{\text{true}}\right)}}.$$

Here, $\mathbf{X}^{\text{true}}$ is the original data matrix (before deleting values) and $\hat{\mathbf{X}}$ is the imputed matrix. With **mean** and **var** we abbreviate the empirical mean and variance, calculated over only the missing entries.

4.1.1 Simulation Study

We consider the MVN model $\sim\mathcal{N}_p(0,\Sigma)$ with

- Model 1: $p = 50$; Σ: block diagonal with 25 blocks of the form $\left(\begin{smallmatrix} 1 & 0.9 \\ 0.9 & 1 \end{smallmatrix}\right)$.
- Model 2: $p = 100$; Σ : two blocks B_1, B_2 each of size 50×50 with $\mathrm{B}_1 = \mathrm{I}_{50}$ and $(\mathrm{B}_2)_{j,j'} = 0.9^{|j-j'|}$.
- Model 3: $p = 55$; Σ : block diagonal with $b = 1, \ldots, 10$ (increasing) blocks B_b of the size $b \times b$, with $(\mathrm{B}_b)_{j,j'} = 0.9$ $(j \neq j')$ and $(\mathrm{B}_b)_{j,j} = 1$.
- Model 4: $p = 500$; $\Sigma_{j,j'} = 0.9^{|j-j'|}$ for $j, j' = 1, \ldots, p$.

For all four settings we perform 50 independent simulation runs. In each run we generate $n = 50$ i.i.d samples from the model. We then delete randomly $10\%, 20\%$ and 30% of the values in the data matrix, apply an imputation method and compute the NRMSE. The results of the different imputation methods (optimally tuned) are reported in Table 2. MissPALasso is very competitive in all setups. SoftImpute works rather poorly, perhaps because the resulting data matrices are not well approximable by low-rank matrices. KnnImpute works very well in model 1 and model 4. Model 1, where each variable is highly correlated with its neighboring variable, represents an example which fits well into the KnnImpute framework. However, in model 2 and model 3, KnnImpute performs rather poorly. The reason is that with an inhomogeneous covariance matrix, as in model 2 and 3, the optimal number of nearest neighbors is varying among the different blocks, and a single parameter K is too restrictive. For example in model 2, variable four is correlated to no other variable, whereas variable twenty-eight is correlated to several other variables. The MissGLasso works well in model 2 and model 3, but significantly worse than the MissPALasso in model 1 and model 4. Arguably, we consider here only multivariate normal models which are ideal, from a distributional point of view, for MissGLasso and our MissPALasso. The more interesting case will be with real data (all from genomics) where model assumptions never hold exactly.

		KnnImpute	SoftImpute	MissGLasso	MissPALasso
Model 1	10%	0.5228 (0.0051)	0.7447 (0.0038)	0.5866 (0.0057)	0.5389 (0.0055)
	20%	0.5948 (0.0052)	0.8149 (0.0033)	0.6822 (0.0046)	0.6152 (0.0044)
	30%	0.6643 (0.0046)	0.8810 (0.0028)	0.7767 (0.0041)	0.6883 (0.0041)
Model 2	10%	0.8502 (0.0049)	0.8619 (0.0032)	0.7909 (0.0039)	0.7846 (0.0038)
	20%	0.8613 (0.0036)	0.8745 (0.0025)	0.8042 (0.0027)	0.7956 (0.0030)
	30%	0.8717 (0.0028)	0.8836 (0.0024)	0.8179 (0.0029)	0.8052 (0.0029)
Model 3	10%	0.4543 (0.0057)	0.4856 (0.0042)	0.4069 (0.0047)	0.4136 (0.0049)
	20%	0.4683 (0.0045)	0.5093 (0.0032)	0.4194 (0.0033)	0.4204 (0.0034)
	30%	0.4937 (0.0037)	0.5447 (0.0035)	0.4451 (0.0036)	0.4388 (0.0034)
Model 4	10%	0.3721 (0.0014)	0.7933 (0.0015)	0.4211 (0.0012)	0.3823 (0.0012)
	20%	0.4139 (0.0011)	0.8157 (0.0013)	0.4548 (0.0012)	0.4038 (0.0011)
	30%	0.4485 (0.0011)	0.8394 (0.0011)	0.4990 (0.0012)	0.4318 (0.0012)

Table 2: Average (SE) NRMSE of KnnImpute, SoftImpute, MissGLasso and MissPALasso with different degrees of missingness.

4.1.2 Real Data Examples

We consider the following four publicly available datasets:

- *Isoprenoid gene network in Arabidopsis thaliana:* The number of genes in the network is $p = 39$. The number of observations (gene expression profiles), corresponding to different experimental conditions, is $n = 118$. More details about the data can be found in Wille *et al.* (2004).
- *Colon cancer:* In this dataset, expression levels of 40 tumor and 22 normal colon tissues ($n = 62$) for $p = 2000$ human genes are measured. For more information see Alon *et al.* (1999).
- *Lymphoma:* This dataset, presented in Alizadeh *et al.* (2000), contains gene expression levels of 42 samples of diffuse large B-cell lymphoma, 9 observations of follicular lymphoma, and 11 cases of chronic lymphocytic leukemia. The total sample size is $n = 62$, and the expression of $p = 4026$ well-measured genes is documented.
- *Yeast cell-cycle:* The dataset, described in Spellman *et al.* (1998), monitors expressions of $p = 6178$ genes. The data consists of four parts, which are relevant to alpha factor (18 samples), elutriation (14 samples), *cdc15* (24 samples), and *cdc28* (17 samples). The total sample size is $n = 73$.

To keep the computational effort reasonable we use for datasets 2-4 only the 100 variables (genes) exhibiting the highest empirical variances. For all datasets we standardize the columns (genes) to zero mean and variance of one. The isoprenoid and the colon cancer dataset contain no missing values. In the reduced lymphoma dataset 5.63% and in the reduced yeast cell-cycle dataset 5.52% of the values are missing. In order to compare the performance of the different imputation methods we delete additional values (randomly) to obtain an overall missing rate of 10%, 20% and 30%. Table 3 shows the results for 50 simulation runs, where in each run another random set of values is deleted.

		KnnImpute	SoftImpute	MissGLasso	MissPALasso
Arabidopsis	10%	0.7723 (0.0073)	0.7222 (0.0052)	0.7237 (0.0064)	0.7178 (0.0062)
	20%	0.8100 (0.0039)	0.7440 (0.0035)	0.7521 (0.0045)	0.7469 (0.0045)
	30%	0.8354 (0.0039)	0.7683 (0.0023)	0.7741 (0.0025)	0.7691 (0.0024)
Colon cancer	10%	0.6371 (0.0034)	0.5703 (0.0024)	0.5674 (0.0031)	0.5634 (0.0026)
	20%	0.6519 (0.0027)	0.5930 (0.0022)	0.5883 (0.0028)	0.5842 (0.0025)
	30%	0.6683 (0.0019)	0.6183 (0.0015)	0.6091 (0.0020)	0.6057 (0.0018)
Lymphoma	10%	0.5090 (0.0052)	0.4936 (0.0039)	0.4127 (0.0047)	0.4034 (0.0045)
	20%	0.5393 (0.0035)	0.5248 (0.0023)	0.4396 (0.0033)	0.4214 (0.0030)
	30%	0.5753 (0.0025)	0.5615 (0.0017)	0.4746 (0.0023)	0.4465 (0.0021)
Yeast cell-cycle	10%	0.5902 (0.0053)	0.4872 (0.0041)	0.4628 (0.0046)	0.4541 (0.0050)
	20%	0.6160 (0.0025)	0.5439 (0.0021)	0.5035 (0.0026)	0.4969 (0.0028)
	30%	0.6473 (0.0021)	0.5592 (0.0017)	0.5317 (0.0021)	0.5272 (0.0022)

Table 3: Average (SE) NRMSE of KnnImpute, SoftImpute, MissGLasso and MissPALasso for different real datasets from genomics.

MissPALasso exhibits in all setups the lowest averaged normalized root mean squared error. The MissGLasso performs nearly as well as the MissPALasso. SoftImpute works well for the arabidopsis and colon cancer datasets but significantly worse for the remaining two datasets. Interestingly, KnnImpute performs for all datasets much inferior to the MissPALasso or the MissGLasso. In light of the simulation results of Section 4.1.1, a reason for the poor performance could be that KnnImpute has difficulties with the inhomogeneous correlation structure between different genes which is plausible to be present in real datasets.

The lymphoma and yeast cell-cycle datasets have "real" missing values. From the left panel of Figures 1 and 2 we can read off how many values are missing for each of the 100 variables. In the right panel of Figures 1 and 2 we show how well the different methods are able to estimate $2\%, 4\%, 6\% \ldots, 16\%$ of artificially deleted entries.

4.2 Computational Efficiency

We compare the MissPA with the standard EM described for example in Schafer (1997). The reason why our algorithm takes less time to converge is because of the more frequent updating of the latent distribution in the M-Steps. A key attribute of the MissPA is that the computational cost of one cycle through all patterns is the same as the cost of a single E-Step of the standard EM. This is a big contrast to the incremental EM, mostly applied to finite mixtures (Thiesson *et al.*, 2001; Ng and McLachlan, 2003), where there is a trade-off between the additional computation time per cycle, or "scan" in the language of Ng and McLachlan (2003), and the fewer number of "scans" required because of the more frequent updating after each partial E-Step. The speed of convergence of the standard EM and the MissPA for three datasets are shown in Figure 3, in which the log-likelihood is plotted as a function of the number of iterations (cycles). The left panel corresponds to the subset of the lymphoma dataset when only the ten genes with highest missing rate are used. This results in a 62×10 data matrix with 22.85% missing values. For the middle panel we draw a random sample of size 62×10 from $\mathcal{N}_{10}(0, \Sigma)$, $\Sigma_{j,j'} = 0.9^{|j-j'|}$, and delete the same entries which are missing in the reduced lymphoma data. For the right panel we draw from the multivariate t-model with degrees of freedom equal to one and again with the same values deleted. As can be seen, the MissPA converges after fewer cycles. A very extreme example is obtained with the multivariate t-model where the standard EM reaches the log-likelihood level of the MissPA about 400 iterations later. We note here, that the result in the right panel highly depends on the realized random sample. With other realizations, we get less and more extreme results than the one shown in Figure 3.

We end this section by illustrating the computational timings of the MissPALasso implemented with the statistical computing language and environment **R**. We consider model 4 of Section 4.1.1 with $n = 50$ and a growing number of variables p ranging from 10 to 500. For each p we delete 10% of the data, run the MissPALasso on a decreasing grid (on the log-scale) of λ values with 30 grid points. For a fixed λ we stop the algorithm if the relative change in imputation satisfies,

$$\frac{\|\hat{\mathbf{X}}^{(r+1)} - \hat{\mathbf{X}}^{(r)}\|^2}{\|\hat{\mathbf{X}}^{(r+1)}\|^2} \leq 10^{-5}.$$

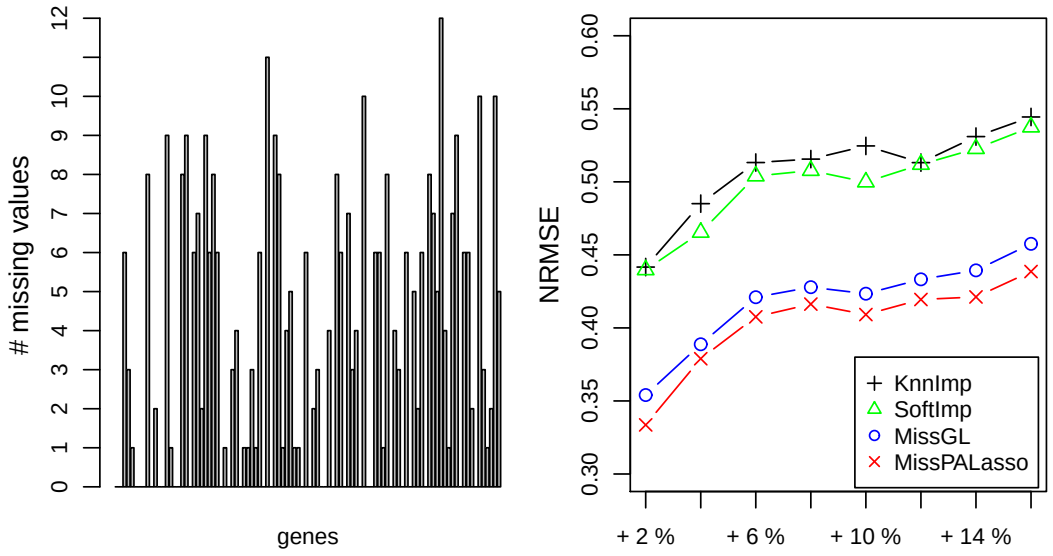

Figure 1: Lymphoma dataset. Left panel: Barplots which count the number of missing values for each of the 100 genes. Right panel: NRMSE for KnnImpute, SoftImpute, MissGLasso and MissPALasso if we introduce additional $2\%, 4\%, 6\%, \ldots, 16\%$ missings.

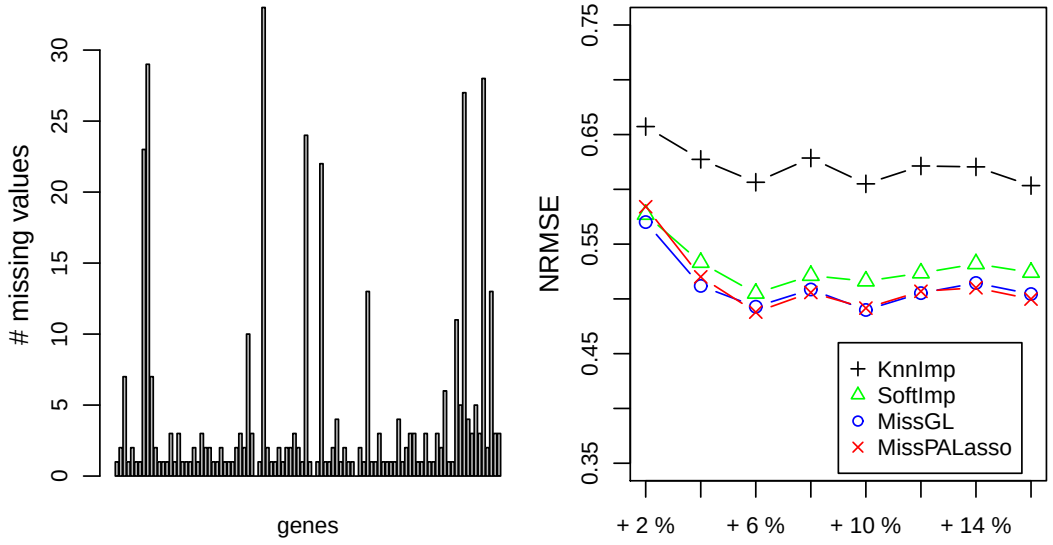

Figure 2: Yeast cell-cycle dataset. Left panel: Barplots which count the number of missing values for each of the 100 genes. Right panel: NRMSE for KnnImpute, SoftImpute, MissGLasso and MissPALasso if we introduce additional $2\%, 4\%, 6\%, \ldots, 16\%$ missings.

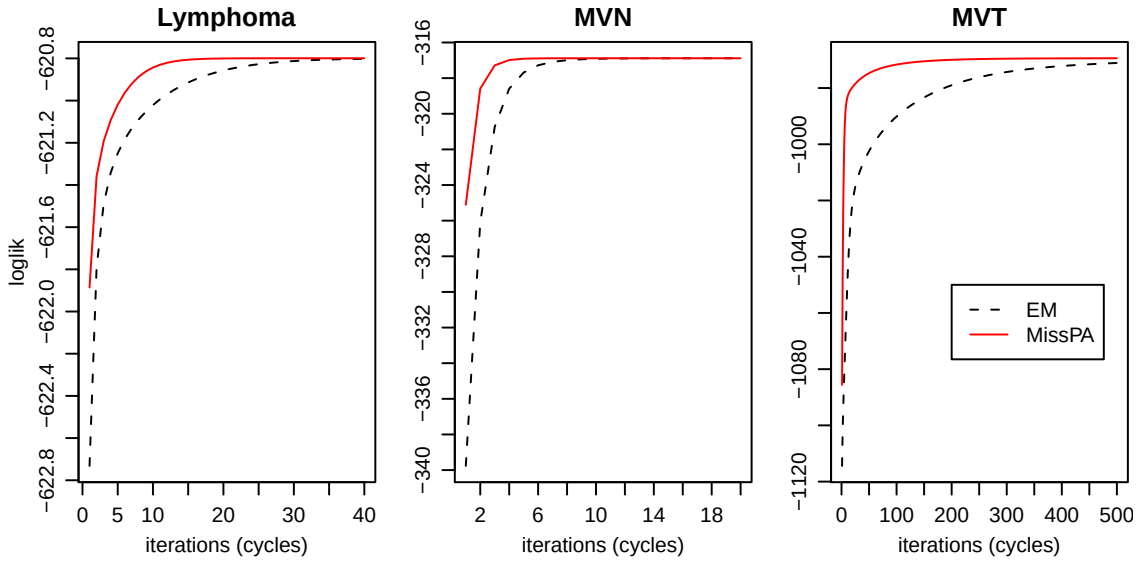

Figure 3: Log-likelihood as a function of the number of iterations (cycles) for the standard EM and the MissPA. Left panel: subset of the lymphoma data ($n = 62$, $p = 10$ and 22.85% missing values). Middle panel: random sample of the size 62×10 from the multivariate normal model with the same missing entries as in the reduced lymphoma data. Right panel: random sample of the size 62×10 from the multivariate t-model again with the same missing values.

In Figure 4 the CPU time in seconds is plotted for various values of p. As shown, we are typically able to solve a problem of size $p = 100$ in about 5 seconds and a problem of size $p = 500$ in about 100 seconds.

5 Convergence Theory for the MissPA algorithm

We will now discuss the numerical properties of the MissPA from Section 3.2. As already mentioned, the pattern alternating maximization algorithm extends the EM in two ways. First, for each pattern a different complete data space is used. Namely, say for pattern k, only those samples are augmented which do not belong to pattern k. Secondly, the E-Step is performed only on those samples belonging to a single pattern.

Fessler and Hero (1994) introduce the SAGE algorithm which generalizes the traditional EM in the sense that it allows different data augmentation schemes. They show monotonicity of the log-likelihood. Nevertheless, the SAGE methodology can not be used here, as generalizations of the E-Step are not allowed in their framework.

On the other hand, Neal and Hinton (1998) give a framework which justifies variants of the EM that allow generalizing the E-Step. One example is the incremental EM, where in each E-Step the expectation is only for one (or a few) samples recalculated before re-estimating the parameters of interest in the following M-Step. The price to pay for such

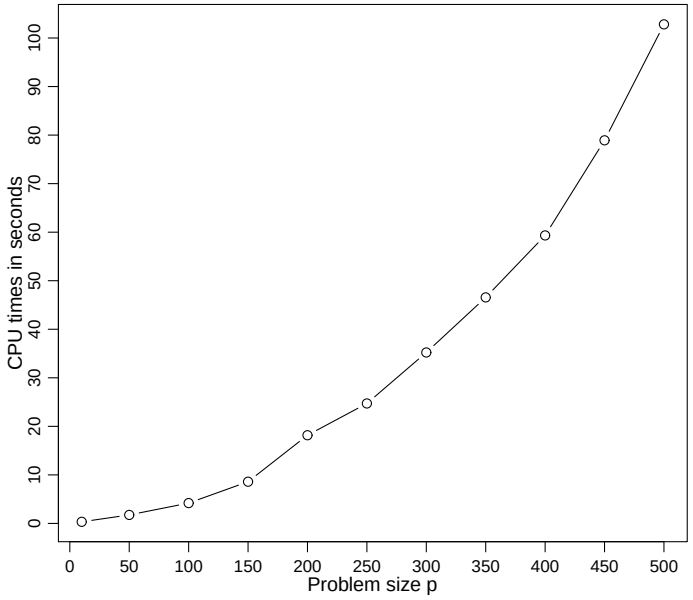

Figure 4: CPU times (in seconds) vs. problem size p of the MissPALasso applied on a grid of thirty λ values. Sample size is $n = 50$.

a generalization is that monotonicity of the log-likelihood is not guaranteed anymore. However, Gunawardana and Byrne (2005) show that such an algorithm still converges to a stationary point of the likelihood. Neal and Hinton (1998) and Gunawardana and Byrne (2005) assume a fixed complete data space in their theory which does not apply to our MissPA.

As outlined, neither the SAGE methodology nor the framework from the incremental EM are suitable to our proposed algorithm. In Section 5.2 we provide a new framework and arguments which justify alternating between complete data spaces *and* partially performing the E-Step. The MissPA will not necessarily increase the log-likelihood in each EM-Step. However, in Section 5.3 we prove convergence to a stationary point of the observed log-likelihood.

Before that, we discuss the convergence properties of two special cases. In a situation with only one missingness pattern ($s = 1$), explicit maximization is possible as discussed in Section 3.1 and therefore, MissPA finds the optimal solution after a single M-Step. Another special case occurs with $s = 2$. With two patterns it is easy to recognize that the E-Step is performed on all augmented data. Thus, the E-Step has not anymore a partial character and the MissPA is a special case of a SAGE algorithm for which monotonicity of the log-likelihood is guaranteed.

5.1 Pattern-Depending Lower Bounds

In order to simplify the notation we assume from now on that all observations have missing values, i.e., for every $i \in \{1, \ldots, n\}$ there is a pattern k such that $i \in \mathcal{I}_k$. Denote the density

$\mathrm{P}_\Sigma(\mathbf{X}^k)$ of $\mathbf{X}^k$, $k \in \{1, \ldots, s\}$, by

$$\mathrm{P}_\Sigma(\mathbf{X}^k) = \prod_{i \in \mathcal{I}_k} p(x_i; \Sigma).$$

Similarly, we define for $k, l \in \{1, \ldots, s\}$

$$\begin{aligned} \mathrm{P}_\Sigma(\mathbf{X}^l_{o_k}) &= \prod_{i \in \mathcal{I}_l} p(x_{i,o_k}; \Sigma_{o_k}) \quad \text{and} \\ \mathrm{P}_\Sigma(\mathbf{X}^l_{m_k} | \mathbf{X}^l_{o_k}) &= \prod_{i \in \mathcal{I}_l} p(x_{i,m_k} | x_{i,o_k}; B_{m_k|o_k}, \Sigma_{m_k|o_k}). \end{aligned}$$

Set $\{\Sigma_l\}_{l \neq k} = (\Sigma_1, \ldots, \Sigma_{k-1}, \Sigma_{k+1}, \ldots, \Sigma_s)$ and consider for $k = 1, \ldots, s$

$$\mathcal{F}_k[\Sigma_k || \{\Sigma_l\}_{l \neq k}] = \log \mathrm{P}_{\Sigma_k}(\mathbf{X}^k_{o_k}) + \sum_{l \neq k} \big(\mathbb{E}_{\Sigma_l}[\log \mathrm{P}_{\Sigma_k}(\mathbf{X}^l) | \mathbf{X}^l_{o_l}] + \mathcal{H}_l[\Sigma_l] \big).$$

Here $\mathcal{H}_l[\tilde{\Sigma}] = -\mathbb{E}_{\tilde{\Sigma}}[\log \mathrm{P}_{\tilde{\Sigma}}(\mathbf{X}^l_{m_l} | \mathbf{X}^l_{o_l}) | \mathbf{X}^l_{o_l}]$ denotes the entropy. Note that $\mathcal{F}_k$ is defined w.r.t fixed observed data $\mathbf{X}_{\text{obs}}$. The subscript k highlights the dependence on the pattern k. Furthermore, for fixed $\mathbf{X}_{\text{obs}}$ and fixed k, $\mathcal{F}_k$ is a function in the parameters $(\Sigma_1, \ldots, \Sigma_s)$.

As a further tool we write the Kullback-Leibler divergence in the following form:

$$\mathcal{D}_l[\tilde{\Sigma} || \Sigma] = \mathbb{E}_{\tilde{\Sigma}}[-\log \big(\mathrm{P}_\Sigma(\mathbf{X}^l_{m_l} | \mathbf{X}^l_{o_l}) / \mathrm{P}_{\tilde{\Sigma}}(\mathbf{X}^l_{m_l} | \mathbf{X}^l_{o_l}) \big) | \mathbf{X}^l_{o_l}]. \tag{5.10}$$

An important property of the Kullback-Leibler divergence is its non-negativity:

$$\begin{aligned} &\mathcal{D}_l[\tilde{\Sigma} || \Sigma] \geq 0, \quad \text{with equality if and only if} \\ &\mathrm{P}_{\tilde{\Sigma}}(\mathbf{X}^l_{m_l} | \mathbf{X}^l_{o_l}) = \mathrm{P}_\Sigma(\mathbf{X}^l_{m_l} | \mathbf{X}^l_{o_l}). \end{aligned}$$

A simple calculation shows that

$$\mathbb{E}_{\tilde{\Sigma}}[\log \mathrm{P}_\Sigma(\mathbf{X}^l) | \mathbf{X}^l_{o_l}] + \mathcal{H}_l[\tilde{\Sigma}] = -\mathcal{D}_l[\tilde{\Sigma} || \Sigma] + \log \mathrm{P}_\Sigma(\mathbf{X}^l_{o_l}). \tag{5.11}$$

Using Equation (5.11), $\mathcal{F}_k[\Sigma_k || \{\Sigma_l\}_{l \neq k}]$ takes the convenient form

$$\mathcal{F}_k[\Sigma_k || \{\Sigma_l\}_{l \neq k}] = \ell(\Sigma_k; \mathbf{X}_{\text{obs}}) - \sum_{l \neq k} \mathcal{D}_l[\Sigma_l || \Sigma_k]. \tag{5.12}$$

In particular, for fixed values of $\{\Sigma_l\}_{l \neq k}$, $\mathcal{F}_k[\,\cdot\,||\{\Sigma_l\}_{l \neq k}]$ lower bounds the observed log-likelihood $\ell(\,\cdot\,; \mathbf{X}_{\text{obs}})$ due to the non-negativity of the Kullback-Leibler divergence.

5.2 MissPA: Optimization Transfer to Pattern-Depending Lower Bounds

We give now an alternative description of the MissPA algorithm. In cycle $r+1$ through all patterns, generate $(\Sigma_1^{r+1}, \ldots, \Sigma_s^{r+1})$ given $(\Sigma_1^r, \ldots, \Sigma_s^r)$ according to

$$\Sigma_k^{r+1} = \arg\max_\Sigma \mathcal{F}_k[\Sigma | \mathrm{Z}_k^{r+1}], \quad k = 1, \ldots, s, \tag{5.13}$$

with $\mathrm{Z}_k^{r+1} = (\Sigma_1^{r+1}, \ldots, \Sigma_{k-1}^{r+1}, \Sigma_{k+1}^{r}, \ldots, \Sigma_s^{r})$.

We have

$$\begin{aligned}\mathcal{F}_k[\Sigma||\mathrm{Z}_k^{r+1}] &= \log \mathrm{P}_\Sigma(\mathbf{X}_{o_k}^k) + \sum_{l<k} \Big(\mathbb{E}_{\Sigma_l^{r+1}}[\log \mathrm{P}_\Sigma(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] + \mathcal{H}_l[\Sigma_l^{r+1}] \Big) \\ &\quad + \sum_{l>k} \Big(\mathbb{E}_{\Sigma_l^{r}}[\log \mathrm{P}_\Sigma(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] + \mathcal{H}_l[\Sigma_l^{r}] \Big).\end{aligned}$$

The entropy terms do not depend on the optimization parameter Σ, therefore,

$$\mathcal{F}_k[\Sigma||\mathrm{Z}_k^{r+1}] = \mathrm{const} + \log \mathrm{P}_\Sigma(\mathbf{X}_{o_k}^k) + \sum_{l<k} \mathbb{E}_{\Sigma_l^{r+1}}[\log \mathrm{P}_\Sigma(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] + \sum_{l>k} \mathbb{E}_{\Sigma_l^{r}}[\log \mathrm{P}_\Sigma(\mathbf{X}^l)|\mathbf{X}_{o_l}^l].$$

Using the factorization $\log \mathrm{P}_\Sigma(\mathbf{X}^l) = \log \mathrm{P}(\mathbf{X}_{o_k}^l; \Sigma_{o_k}) + \log \mathrm{P}(\mathbf{X}_{m_k}^l | \mathbf{X}_{o_k}^l; B_{m_k|o_k}, \Sigma_{m_k|o_k})$ (for all $l \neq k$), and separate maximization w.r.t Σ_{o_k} and $(B_{m_k|o_k}, \Sigma_{m_k|o_k})$ we end up with the expressions from the M-Step of the MissPA. Summarizing the above, we have recovered the M-Step as a maximization of $\mathcal{F}_k[\Sigma||\mathrm{Z}_k^{r+1}]$ which is a lower bound of the observed log-likelihood. Or in the language of Lange *et al.* (2000), optimization of $\ell(\,\cdot\,; \mathbf{X}_{\mathrm{obs}})$ is transferred to the surrogate objective $\mathcal{F}_k[\,\cdot\,||\mathrm{Z}_k^{r+1}]$.

There is still an important piece missing: In M-Step k of cycle $r+1$ we are maximizing $\mathcal{F}_k[\,\cdot\,||\mathrm{Z}_k^{r+1}]$ whereas in the following M-Step $(k+1)$ we optimize $\mathcal{F}_{k+1}[\,\cdot\,||\mathrm{Z}_{k+1}^{r+1}]$. In order for the algorithm to make progress, it is essential that $\mathcal{F}_{k+1}[\,\cdot\,||\mathrm{Z}_{k+1}^{r+1}]$ attains higher values than its predecessor $\mathcal{F}_k[\,\cdot\,||\mathrm{Z}_k^{r+1}]$. In this sense the following Proposition is crucial.

Proposition 5.1. *For $r = 0, 1, 2, \ldots$ we have that*

$$\begin{aligned}&\mathcal{F}_s[\Sigma_s^r||\mathrm{Z}_s^r] \leq \mathcal{F}_1[\Sigma_s^r||\mathrm{Z}_1^{r+1}], \quad \textit{and} \\ &\mathcal{F}_k[\Sigma_k^{r+1}||\mathrm{Z}_k^{r+1}] \leq \mathcal{F}_{k+1}[\Sigma_k^{r+1}||\mathrm{Z}_{k+1}^{r+1}] \quad \textit{for } k = 1, \ldots, s-1.\end{aligned}$$

Proof. We have,

$$\mathcal{F}_k[\Sigma_k^{r+1}||\mathrm{Z}_k^{r+1}] = \log \mathrm{P}_{\Sigma_k^{r+1}}(\mathbf{X}_{o_k}^k) + \mathbb{E}_{\Sigma_{k+1}^r}[\log \mathrm{P}_{\Sigma_k^{r+1}}(\mathbf{X}^{k+1})|\mathbf{X}_{o_{k+1}}^{k+1}] + \mathcal{H}_{k+1}[\Sigma_{k+1}^r] + \mathrm{rest}$$

and

$$\mathcal{F}_{k+1}[\Sigma_k^{r+1}||\mathrm{Z}_{k+1}^{r+1}] = \log \mathrm{P}_{\Sigma_k^{r+1}}(\mathbf{X}_{o_{k+1}}^{k+1}) + \mathbb{E}_{\Sigma_k^{r+1}}[\log \mathrm{P}_{\Sigma_k^{r+1}}(\mathbf{X}^{k})|\mathbf{X}_{o_k}^{k}] + \mathcal{H}_{k}[\Sigma_k^{r+1}] + \mathrm{rest}$$

where

$$\mathrm{rest} = \sum_{l<k} \big(\mathbb{E}_{\Sigma_l^{r+1}}[\log \mathrm{P}_{\Sigma_k^{r+1}}(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] + \mathcal{H}_l[\Sigma_l^{r+1}] \big) + \sum_{l>k+1} \big(\mathbb{E}_{\Sigma_l^{r}}[\log \mathrm{P}_{\Sigma_k^{r+1}}(\mathbf{X}^l)|\mathbf{X}_{o_l}^l] + \mathcal{H}_l[\Sigma_l^{r}] \big).$$

Furthermore, using (5.11) and noting that $\mathcal{D}_k[\Sigma_k^{r+1}||\Sigma_k^{r+1}] = 0$, we obtain

$$\begin{aligned}\mathcal{F}_k[\Sigma_k^{r+1}||\mathrm{Z}_k^{r+1}] - \mathcal{F}_{k+1}[\Sigma_k^{r+1}||\mathrm{Z}_{k+1}^{r+1}] &= \mathcal{D}_k[\Sigma_k^{r+1}||\Sigma_k^{r+1}] - \mathcal{D}_{k+1}[\Sigma_{k+1}^{r}||\Sigma_k^{r+1}] \\ &= -\mathcal{D}_{k+1}[\Sigma_{k+1}^{r}||\Sigma_k^{r+1}] \leq 0.\end{aligned}$$

Note that equality holds if and only if $\mathrm{P}_{\Sigma_k^{r+1}}(\mathbf{X}_{m_{k+1}}^{k+1}|\mathbf{X}_{o_{k+1}}^{k+1}) = \mathrm{P}_{\Sigma_{k+1}^{r}}(\mathbf{X}_{m_{k+1}}^{k+1}|\mathbf{X}_{o_{k+1}}^{k+1})$. □

In light of Proposition 5.1 it is clear that (5.13) generates a monotonely increasing sequence of the form:

$$\begin{aligned}&\mathcal{F}_s[\Sigma_s^0||\mathrm{Z}_s^0] \leq \mathcal{F}_1[\Sigma_s^0||\mathrm{Z}_1^1] \leq \mathcal{F}_1[\Sigma_1^1||\mathrm{Z}_1^1] \leq \mathcal{F}_2[\Sigma_1^1||\mathrm{Z}_2^1] \leq \mathcal{F}_2[\Sigma_2^1||\mathrm{Z}_2^1] \leq \cdots \\ &\cdots \leq \mathcal{F}_k[\Sigma_k^{r+1}||\mathrm{Z}_k^{r+1}] \leq \mathcal{F}_{k+1}[\Sigma_k^{r+1}||\mathrm{Z}_{k+1}^{r+1}] \leq \mathcal{F}_{k+1}[\Sigma_{k+1}^{r+1}||\mathrm{Z}_{k+1}^{r+1}] \leq \cdots\end{aligned}$$

For example, we can deduce that $\{\mathcal{F}_s[\Sigma_s^r||\mathrm{Z}_s^r]\}_{r=0,1,2,\ldots}$ is a monotone increasing sequence in r.

5.3 Convergence to Stationary Points

Ideally we would like to show that a limit point of the sequence generated by the MissPA algorithm is a global maximum of $\ell(\Sigma; \mathbf{X}_{\mathrm{obs}})$. Unfortunately, this is too ambitious because for general missing data patterns the observed log-likelihood is a non-concave function with several local maxima. Thus, the most we can expect is that our algorithm converges to a stationary point. This is ensured by the following theorem which is proved in the Appendix.

Theorem 5.1. *Assume that* $\mathcal{K} = \{(\Sigma_1, \ldots, \Sigma_s) : \mathcal{F}_s[\Sigma_s||\Sigma_1, \ldots, \Sigma_{s-1}] \geq \mathcal{F}_s[\Sigma_s^0||\mathrm{Z}_s^0]\}$ *is compact. Then every limit point* $\bar{\Sigma}_s$ *of* $\{\Sigma_s^r\}_{r=0,1,2,\ldots}$ *is a stationary point of* $\ell(\,\cdot\,; \mathbf{X}_{\mathrm{obs}})$.

6 Conclusions

We have presented a new algorithm, the MissPA, for maximizing the observed log-likelihood for a multivariate normal data matrix with missing values. Simplified, our algorithm iteratively cycles through the different missingness patterns, performs multivariate regressions of the missing on the observed variables and uses these regression coefficients for partial imputation of the missing values. We argued theoretically and gave numerical examples showing that the MissPA is computationally more efficient than the standard EM algorithm. Furthermore, we analyze the numerical properties using non-standard arguments and prove that solutions of the MissPA converge to stationary points of the observed log-likelihood.

In a high-dimensional setup with $p \gg n$ the regression interpretation of the MissPA opens up the door to do regularization by replacing least squares regressions with Lasso analogues. Our proposed algorithm, the MissPALasso, performs a coordinate descent approximation of the corresponding Lasso problem in order to gain speed. On simulated

and four real datasets (all from genomics) we demonstrate that the MissPALasso outperforms alternative imputation techniques such as k-nearest neighbors imputation, nuclear norm minimization or a penalized likelihood approach with an ℓ_1-penalty on the inverse covariance matrix.

A Proof of Theorem 5.1

Proof. First, note that the sequence $\{(\Sigma_1^r, \ldots, \Sigma_s^r)\}_{r=0,1,2,\ldots}$ lies in the compact set $\mathcal{K}$. Now, let $\Sigma_s^{r_j}$ be a subsequence converging to $\bar{\Sigma}_s$ as $j \to \infty$. By invoking compactness, we can assume w.l.o.g (by restricting to a subsequence) that $(\Sigma_1^{r_j}, \ldots, \Sigma_s^{r_j}) \to (\bar{\Sigma}_1, \ldots, \bar{\Sigma}_s)$.

As a direct consequence of the monotonicity of the sequence $\{\mathcal{F}_s[\Sigma_s^r||\mathrm{Z}_s^r]\}_{r=0,1,2,\ldots}$ we obtain

$$\lim_r \mathcal{F}_s[\Sigma_s^r||\mathrm{Z}_s^r] = \mathcal{F}_s[\bar{\Sigma}_s||\bar{\Sigma}_1, \ldots, \bar{\Sigma}_{s-1}] \equiv \bar{\mathcal{F}}.$$

From (5.13) and Proposition 5.1, for $k = 1, \ldots, s-1$ and $r = 0, 1, 2, \ldots$, the following "sandwich"-formulae hold:

$$\begin{aligned}
&\mathcal{F}_s[\Sigma_s^r||\mathrm{Z}_s^r] \leq \mathcal{F}_1[\Sigma_s^r||\mathrm{Z}_1^{r+1}] \leq \mathcal{F}_1[\Sigma_1^{r+1}||\mathrm{Z}_1^{r+1}] \leq \mathcal{F}_s[\Sigma_s^{r+1}||\mathrm{Z}_s^{r+1}],\\
&\mathcal{F}_s[\Sigma_s^r||\mathrm{Z}_s^r] \leq \mathcal{F}_{k+1}[\Sigma_k^{r+1}||\mathrm{Z}_{k+1}^{r+1}] \leq \mathcal{F}_{k+1}[\Sigma_{k+1}^{r+1}||\mathrm{Z}_{k+1}^{r+1}] \leq \mathcal{F}_s[\Sigma_s^{r+1}||\mathrm{Z}_s^{r+1}].
\end{aligned}$$

As a consequence we have for $k = 1, \ldots, s-1$

$$\lim_r \mathcal{F}_1[\Sigma_s^r||\mathrm{Z}_1^{r+1}] = \lim_r \mathcal{F}_1[\Sigma_1^{r+1}||\mathrm{Z}_1^{r+1}] = \bar{\mathcal{F}} \quad \text{and} \tag{A.14}$$

$$\lim_r \mathcal{F}_{k+1}[\Sigma_k^{r+1}||\mathrm{Z}_{k+1}^{r+1}] = \lim_r \mathcal{F}_{k+1}[\Sigma_{k+1}^{r+1}||\mathrm{Z}_{k+1}^{r+1}] = \bar{\mathcal{F}}. \tag{A.15}$$

Now consider the sequence $(\Sigma_1^{r_j+1}, \ldots, \Sigma_s^{r_j+1})$. By compactness of $\mathcal{K}$ this sequence converges w.l.o.g to some $(\Sigma_1^*, \ldots, \Sigma_s^*)$. We now show by induction that

$$\bar{\Sigma}_s = \Sigma_1^* = \ldots = \Sigma_s^*.$$

From the 1st M-Step of cycle $r_j + 1$ we have

$$\mathcal{F}_1[\Sigma_1^{r_j+1}||\mathrm{Z}_1^{r_j+1}] \geq \mathcal{F}_1[\Sigma||\mathrm{Z}_1^{r_j+1}] \quad \text{for all} \quad \Sigma.$$

Taking the limit $j \to \infty$ we get:

$$\mathcal{F}_1[\Sigma_1^*||\{\bar{\Sigma}_l\}_{l>1}] \geq \mathcal{F}_1[\Sigma|\{\bar{\Sigma}_l\}_{l>1}] \quad \text{for all} \quad \Sigma.$$

In particular, Σ_1^* is the (unique) maximizer of $\mathcal{F}_1[\cdot||\{\bar{\Sigma}_l\}_{l>1}]$. Assuming $\Sigma_1^* \neq \bar{\Sigma}_s$ would imply

$$\mathcal{F}_1[\Sigma_1^*||\{\bar{\Sigma}_l\}_{l>1}] > \mathcal{F}_1[\bar{\Sigma}_s||\{\bar{\Sigma}_l\}_{l>1}].$$

But this contradicts $\mathcal{F}_1[\Sigma_1^*||\{\bar{\Sigma}_l\}_{l>1}] = \mathcal{F}_1[\bar{\Sigma}_s||\{\bar{\Sigma}_l\}_{l>1}] = \bar{\mathcal{F}}$, which holds by (A.14). Therefore we obtain $\Sigma_1^* = \bar{\Sigma}_s$.

Assume that we have proven $\Sigma_1^* = \ldots = \Sigma_k^* = \bar{\Sigma}_s$. We will show that $\Sigma_{k+1}^* = \bar{\Sigma}_s$. From the k+1st M-Step in cycle $r_j + 1$:

$$\mathcal{F}_{k+1}[\Sigma_{k+1}^{r_j+1} || \mathrm{Z}_{k+1}^{r_j+1}] \geq \mathcal{F}_{k+1}[\Sigma || \mathrm{Z}_{k+1}^{r_j+1}] \quad \text{for all} \quad \Sigma.$$

Taking the limit for $j \to \infty$, we conclude that Σ_{k+1}^* is the (unique) maximizer of

$$\mathcal{F}_{k+1}[\cdot || \{\Sigma_l^*\}_{l<k+1}, \{\bar{\Sigma}_l\}_{l>k+1}].$$

From (A.15),

$$\mathcal{F}_{k+1}[\Sigma_{k+1}^* || \{\Sigma_l^*\}_{l<k+1}, \{\bar{\Sigma}_l\}_{l>k+1}] = \mathcal{F}_{k+1}[\Sigma_k^* || \{\Sigma_l^*\}_{l<k+1}, \{\bar{\Sigma}_l\}_{l>k+1}] = \bar{\mathcal{F}},$$

and therefore Σ_{k+1}^* must be equal to Σ_k^*. By induction we have $\Sigma_k^* = \bar{\Sigma}_s$ and so we proved that $\Sigma_{k+1}^* = \bar{\Sigma}_s$ holds.

Finally, we show stationarity of $\bar{\Sigma}_s$. Invoking (5.12) we can write

$$\mathcal{F}_s[\Sigma || \bar{\Sigma}_s, \ldots, \bar{\Sigma}_s] = \ell(\Sigma; \mathbf{X}_{\text{obs}}) - \sum_{l=1}^{s-1} \mathcal{D}_l[\bar{\Sigma}_s || \Sigma].$$

Note that

$$\frac{\partial}{\partial \Sigma} \mathcal{D}_l[\bar{\Sigma}_s || \Sigma] \Big|_{\bar{\Sigma}_s} = 0.$$

Furthermore, as $\Sigma_s^{r_j+1}$ maximizes $\mathcal{F}_s[\Sigma || \Sigma_1^{r_j+1}, \ldots, \Sigma_{s-1}^{r_j+1}]$, we get in the limit as $j \to \infty$

$$\frac{\partial}{\partial \Sigma} \mathcal{F}_s[\Sigma | \bar{\Sigma}_s, \ldots, \bar{\Sigma}_s] \Big|_{\bar{\Sigma}_s} = \frac{\partial}{\partial \Sigma} \mathcal{F}_s[\Sigma || \Sigma_1^*, \ldots, \Sigma_{s-1}^*] \Big|_{\Sigma_s^*} = 0.$$

Therefore, we conclude that $\frac{\partial}{\partial \Sigma} \ell(\Sigma; \mathbf{X}_{\text{obs}}) \Big|_{\bar{\Sigma}_s} = 0.$ □

References

Aittokallio, T. (2010) Dealing with missing values in large-scale studies: microarray data imputation and beyond. *Briefings in Bioinformatics*, **11**, 253–264.

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O. and Staudt, L. M. (2000) Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Allen, G. I. and Tibshirani, R. (2010) Transposable regularized covariance models with an application to missing data imputation. *arXiv.org: 0906.3465*.

Alon, U., Barkai, N., Notterman, D. A., Gishdagger, K., Ybarradagger, S., Mackdagger, D. and Levine, A. J. (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, **96**, 6745–6750.

Cai, J.-F., Candès, E. J. and Shen, Z. (2008) A singular value thresholding algorithm for matrix completion. *arXiv.org: 0810.3286*.

Candès, E. J. and Recht, B. (2009) Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, **9**, 717–772.

Candès, E. J. and Tao, T. (2009) The power of convex relaxation: Near-optimal matrix completion. *arXiv.org: 0903.1476*.

Dempster, A., Laird, N. and Rubin, D. (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **39**, 1–38.

Fessler, J. A. and Hero, A. O. (1994) Space-alternating generalized Expectation-Maximization algorithm. *IEEE Transactions on Signal Processing*, **42**, 2664–2677.

Friedman, J., Hastie, T. and Tibshirani, R. (2007) Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 432–441.

Friedman, J., Hastie, T. and Tibshirani, R. (2008) Regularized paths for generalized linear models via coordinate descent. Tech. rep., Department of Statistics, Stanford University.

Gunawardana, A. and Byrne, W. (2005) Convergence theorems for generalized alternating minimization procedures. *Journal of Machine Learning Research*, **6**, 2049–2073.

Keshavan, R. H., Oh, S. and Montanari, A. (2009) Matrix completion from a few entries. *arXiv.org: 0901.3150*.

Kim, H., Golub, G. H. and Park, H. (2006) Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, **22**, 1410–1411.

Lange, K., Hunter, D. R. and Yang, I. (2000) Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, **9**, 1–20.

Little, R. J. A. and Rubin, D. (1987) *Statistical Analysis with Missing Data*. Series in Probability and Mathematical Statistics, Wiley.

Mazumder, R., Tibshirani, R. and Hastie, T. (2009) Spectral regularization algorithms for learning large incomplete matrices. Tech. rep.

Neal, R. M. and Hinton, G. E. (1998) A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, 355–368. Kluwer Academic Publishers.

Ng, S. K. and McLachlan, G. J. (2003) On the choice of the number of blocks with the incremental EM algorithm for the fitting of normal mixtures. *Statistics and Computing*, **13**, 45–55.

Oba, S., Sato, M.-A., Takemasa, I., Monden, M., Matsubara, K.-I. and Ishii, S. (2003) A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, **19**, 2088–2096.

Schafer, J. L. (1997) *Analysis of Incomplete Multivariate Data*. Monographs on Statistics and Applied Probability 72, Chapman and Hall.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D. and Futcher, B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, **9**, 3273–97.

Spirtes, P., Glymour, C. and Scheines, R. (2000) *Causation, Prediction, and Search*. The MIT Press, 2nd edition.

Städler, N. and Bühlmann, P. (2010) Missing values: sparse inverse covariance estimation and an extension to sparse regression. *arXiv.org: 0903.5463*.

Thiesson, B., Meek, C. and Heckerman, D. (2001) Accelerating EM for large databases. *Machine Learning*, **45**, 279–299.

Tibshirani, R. (1996) Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, **58**, 267–288.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman, R. B. (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.

Wille, A., Zimmermann, P., Vranova, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., Prelic, A., von Rohr, P., Thiele, L., Zitzler, E., Gruissem, W. and Bühlmann, P. (2004) Sparse graphical Gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. *Genome Biology*, **5**.